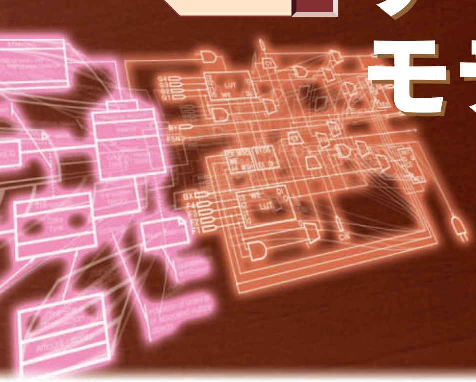
第2章

# UMLでディジタル・カメラをモデリング

## ── 手順を踏んで，要求仕様から設計モデルを導き出す



塚田雄一

**ここではUML（unified modeling language）を使って，要求仕様から設計モデルを導出する手順を説明する．本稿でモチーフとした題材はディジタル・カメラのシステムである．UMLを利用するモデリング手法は，大規模ソフトウェア開発などの世界で実績がある．手順を踏んできちんと分析・モデリングを行えば，適切な設計仕様が得られる．（編集部）**

「ASIC開発やシステムLSI開発において，UMLでどのようにモデリングをしたらよいのか？」という質問をよく受けます．そこでUMTP（UMLモデリング推進協議会）のSoC分科会では，ディジタル・カメラ用のシステムLSIをイメージして，UMLのリファレンス・モデルを作成しています．ディジタル・カメラを題材として選んだのは，画像処理部分があり，また，実際にこの用途でシステムLSI（大規模ASICや大規模ASSP）がよく開発されているためです．

本分科会は，モデリングの要求仕様からハードウェア・ソフトウェア分割を行い，LSI（ハードウェア）化を念頭においてモデリングするところまでを目標としています．現在はモデリングの途上であり，モデルはまだ完成していませんが，本稿ではその一部を紹介します．具体的には，ディジタル・カメラの要求仕様の策定，ドメイン分け，ユースケース分析，分析モデリング，SystemCモデリングについて説明します．

なお，本モデルは実際の製品開発に適用するためのものではなく，あくまでもUMLによるモデリングのサンプルを示すためのものです．また，筆者も含めて今回のモデリングに参加したメンバは，実際に業務でディジタル・カメラ用LSIを開発している技術者ではありません．そのため，要求や機能仕様を簡略化しているところがあります（製品レベルの機能とは異なる部分がある）．あしからずご了承ください．

### ● 五つの工程に分けて作業を進める

まず，開発フローについて説明します．UMLの開発フ

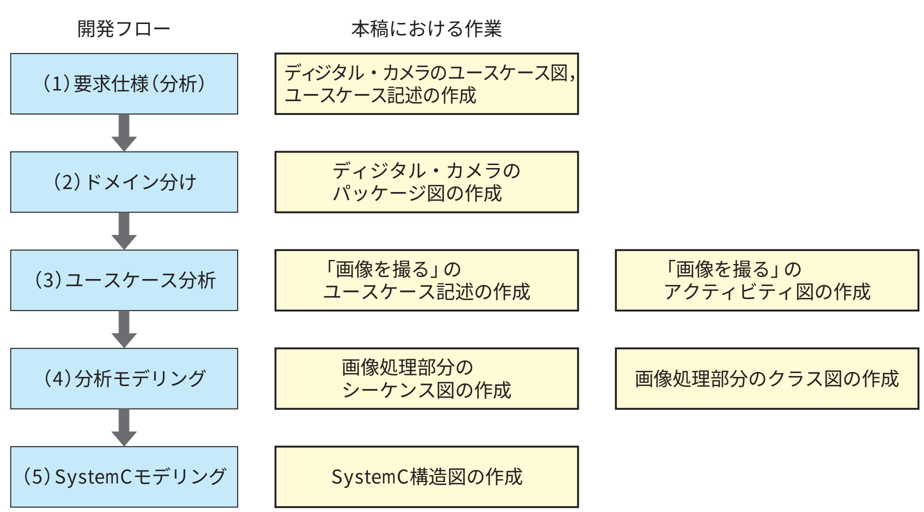



**図1 開発フロー**
今回のモデリングの開発フローを示している．なお，UMLの開発フローについては決まりがあるわけではない．

2

**表1　各工程の作業の概要**

| 開発工程 | 入　力 | 出力(成果物) | 工程の概要 |
|---|---|---|---|
| (1) 要求仕様(分析) | ● 機能要件(要求) | ● (1-1) ユースケース図 | システムが外部に提供する機能(サービス)，および機能(サービス)の提供先を明確化する |
| | | ● (1-2) ユースケース記述またはアクティビティ図 | 各ユースケースに対して，システムのシナリオを記載する |
| (2) ドメイン分け | ● 機能要件(要求)<br>● (1-1) ユースケース図<br>● (1-2) ユースケース記述 | ● (2-1) パッケージ図 | 各機能をドメイン(役割の種別)に分割する |
| (3) ユースケース分析 | ● (1-1) ユースケース図 | ● (3-1) ユースケース図 | 各ユースケースの詳細を抽出する |
| | | ● (3-2) ユースケース記述またはアクティビティ図 | ユースケースごとにシナリオを作る |
| (4) 分析モデリング | ● (3-1) ユースケース図<br>● (3-2) ユースケース記述 | ● (4-1) シーケンス図 | 機能分析を行い，処理の流れをモデリングする |
| | | ● (4-2) 分析クラス図 | 処理の流れをもとにオブジェクト候補をまとめてクラス化する |
| (5) SystemC モデリング | ● (4-2) 分析クラス図 | ● (5-1) SystemC 構造図 | クラス図をSystemC構造図に変換する |

ローについては決まりがあるわけではありません．今回は，**図1**のようなフローでモデリングを行います．また，各工程の作業の概要を**表1**にまとめました．

まず，「要求仕様(分析)」の工程で，ディジタル・カメラの要求分析を行います．ここでは，ユースケース図やユースケース記述などを使用して，システムが外部に提供する機能(サービス)，および機能(サービス)の提供先を明確にします．また，各ユースケースに対して，システムのシナリオを記載します．本稿の例では，ディジタル・カメラのユースケース図とユースケース記述を作成します．

次に，ドメイン分けを行います．ドメイン分けでは，パッケージ図を使用して各機能をドメイン(役割の種別)に分割します．本例では，ディジタル・カメラのパッケージ図を記述しています．

続いて，ユースケース分析の工程です．ユースケース図の中の各ドメインに対してユースケースの詳細を記述します．そして，ユースケースごとにシナリオを作成します．本例では画像を撮る部分のユースケース記述を作成し，さらにアクティビティ図を作成しました．

分析モデリングの工程ではユースケース図，ユースケース記述を作成し，処理の流れを考えるためのシーケンス図を作成します．そして，そのシーケンス図をもとにオブジェクトを抽出し，オブジェクト間の関連を記載します．つまり，分析モデルのクラス図（オブジェクト図）を作成します．本例では，パッケージ図の画像処理の部分について，分析モデルのシーケンス図とクラス図を作成しています．

そして最後に，参考のためのSystemC構造図を記します．**図2**に今回の開発の全体像をまとめました．

### ● ディジタル・カメラのユースケース図を作成する

では，各工程の作業を順番に説明していきます．

まずはじめに，ディジタル・カメラのユースケース図について考えてみましょう(**図3**)．ここではディジタル・カメラの基本的な動作が何であるかを考えます．また，ディジタル・カメラについて，使用者(アクタ)から見てどのような機能があるかということを考えます．

ここでのアクタは撮影者になります．撮影者が何を行うか？ を考えます．撮影者の行動としては，「画像を撮る」，「画像を加工する」，「画像を消去する」，「画像を見る」，「画像を転送する」が考えられます．そして，画像を撮る際には構図を決める必要があるため，「画像を撮る」というユースケースは「構図を決める」というユースケースを含み(include)ます．

最初から上記項目が思い浮かび，すんなり決まったわけではありません．例えば「画像を加工する」という記述がほんとうに必要かどうかを検討しました．また，ディジタル・カメラには撮影画像を確認するためのプレビュー機能などがあるため，「画像を見る」という機能を後から追加しました．つまり，撮影者からどのような要求があるのか？ということを何度も検討する必要があるのです．

### ● 「画像を撮る」のユースケース記述を作成する

次に，ディジタル・カメラのユースケース図のうち，「画像を撮る」の部分のユースケース記述を作成します(**図4**)．

主アクタは撮影者になります．ここでまず，前提条件を考えてみましょう．画像を撮るためには初期化が終わってなければならないため，前提条件は「初期化が終わってい

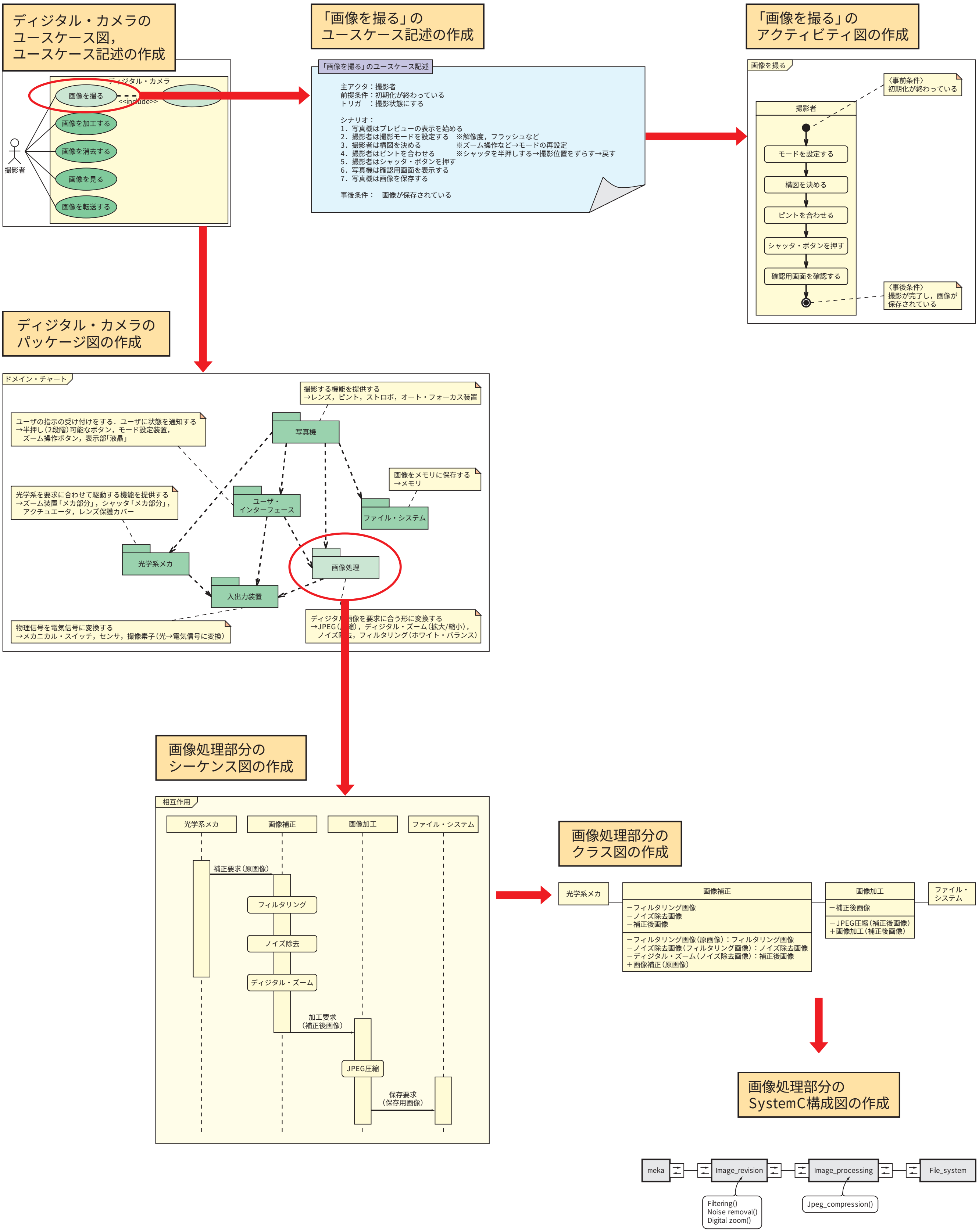



**図2　開発モデル**

**ここでは，ディジタル・カメラのユースケース図，「画像を撮る」のユースケース記述，「画像を撮る」のアクティビティ図，ディジタル・カメラのパッケージ図，画像処理部分のシーケンス図，画像処理部分のクラス図，画像処理部分のSystemC構造図について説明する．**

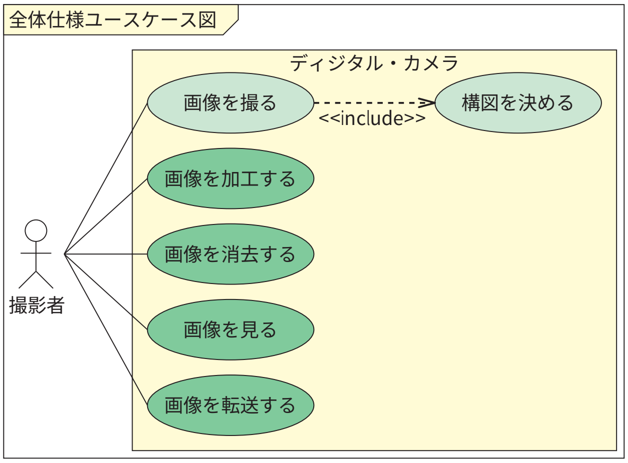



**図3　ディジタル・カメラのユースケース図**

**ディジタル・カメラのユースケースを考えた場合，「画像を撮る」，「画像を加工する」，「画像を消去する」，「画像を見る」，「画像を転送する」が考えられる．**

る」になります．そして，トリガとして「撮影状態にする」ということになります．

撮影状態になると，以下のシナリオが実行されます．すなわち，「写真機はプレビューの表示を始める」→「撮影者は撮影モードを設定する」→「撮影者は構図を決める」→「撮影者はピントを合わせる」→「撮影者はシャッタ・ボタンを押す」→「写真機は確認用画面を表示する」→「写真機は画像を保存する」のシーケンスです．そして事後条件として，「画像が保存されている」ということになります．

ここでは，シナリオにはかならず主語を記述するということがポイントになります．ここでの主語は「撮影者」と「写真機」の二つです．当初，筆者らは主語を記述していませんでしたが，主語を明確にすることによって，よりシナリオが明確に記述できるようになりました．

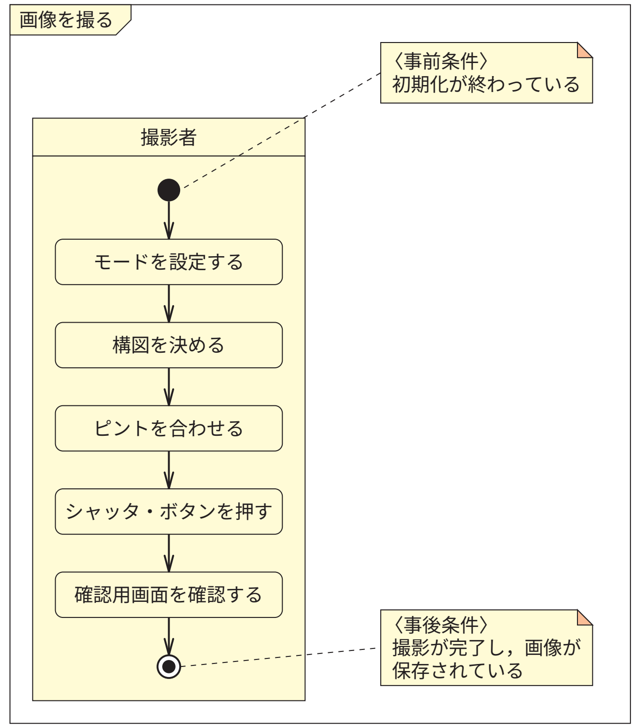



**図5　「画像を撮る」のアクティビティ図**

**「画像を撮る」のアクティビティ図では，「モードを設定する」→「構図を決める」→「ピントを合わせる」→「シャッタ・ボタンを押す」→「確認用画像を確認する」といった処理の流れを書く．**

### ●「画像を撮る」のアクティビティ図を作成する

続いて，アクティビティ図について考えてみましょう．ここでは「画像を撮る」という部分のアクティビティ図を作成します（**図5**）．アクティビティ図は処理の流れを記述するものなので，画像を撮る際の処理の流れを考えます．すなわち，事前条件として「初期化が終わっている」というところから，事後条件として「撮影が完了し，画像が保存さ

「画像を撮る」のユースケース記述

主アクタ：撮影者<br>
前提条件：初期化が終わっている<br>
トリガ　：撮影状態にする

シナリオ：<br>
1．写真機はプレビューの表示を始める<br>
2．撮影者は撮影モードを設定する　※解像度，フラッシュなど<br>
3．撮影者は構図を決める　※ズーム操作など→モードの再設定<br>
4．撮影者はピントを合わせる　※シャッタを半押しする→撮影位置をずらす→戻す<br>
5．撮影者はシャッタ・ボタンを押す<br>
6．写真機は確認用画面を表示する<br>
7．写真機は画像を保存する

事後条件：　画像が保存されている

**図4**
**「画像を撮る」のユースケース記述**

**ユースケース記述はフォーマットが決まっているわけではない．ここでは主アクタ，前提条件，トリガ，シナリオ，事後条件などを記述している．**

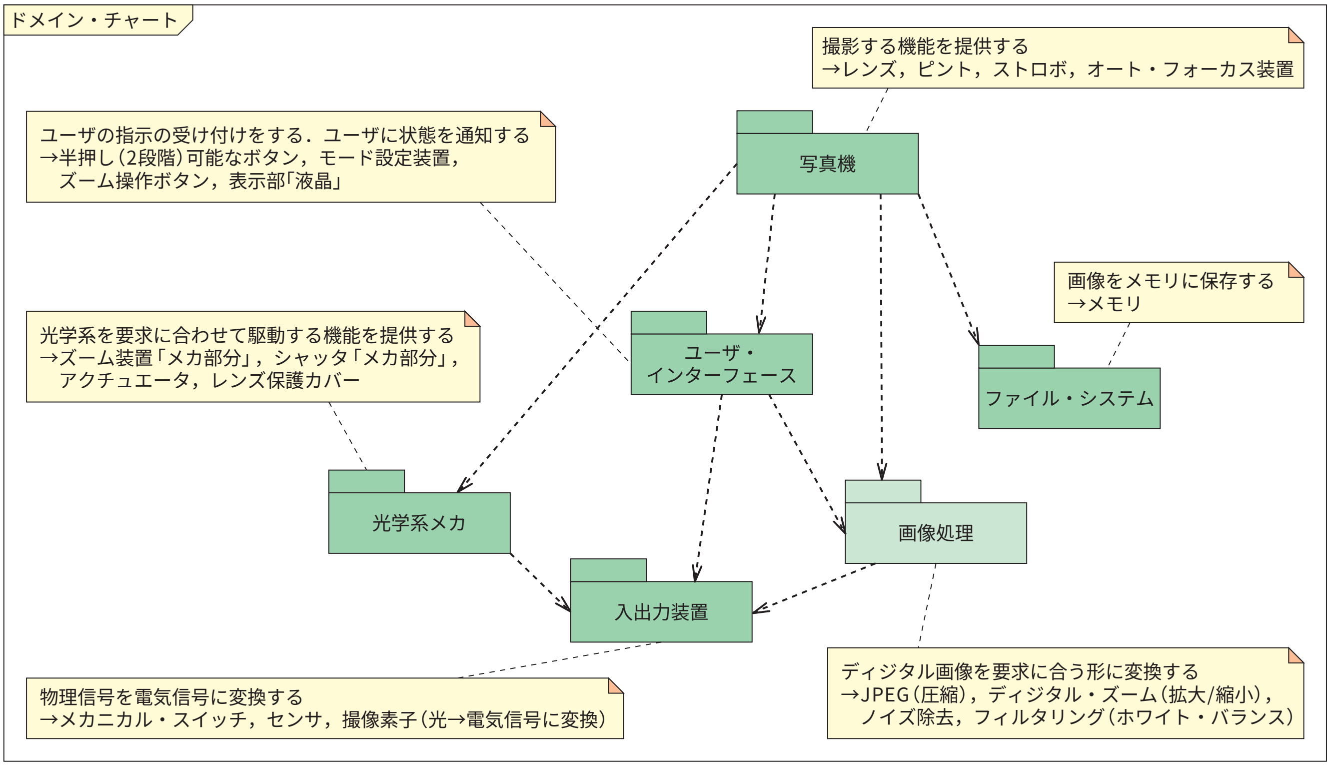



**図6　ディジタル・カメラのパッケージ図**
ディジタル・カメラのパッケージ図には，写真機，ユーザ・インターフェース，光学系メカ，入出力装置，画像処理，ファイル・システムなどが存在する．



れている」までの処理の流れを記述します．

アクタは撮影者で，事前条件と事後条件はユースケース記述と同じになります．そして，「モードを設定する」，「構図を決める」，「ピントを合わせる」，「シャッタ・ボタンを押す」，「確認用画像を確認する」という処理の流れになります．

アクティビティ図は，フローチャートを記述する感覚で作成するとよいでしょう．ただし，詳細まで記述せずに，処理の概要のみを記述することがたいせつです．目安としては，多くても1ページに記述できる程度(処理が5個～20個)にしたほうがよいと思います．

### ● ディジタル・カメラのパッケージ図を作成する

次に，ディジタル・カメラのドメインを考えます(**図6**)．ここではディジタル・カメラがどのような要素(ドメイン)から成り立っているのかを記述します．ユースケース図によって要求を分析し，その要求を満足するための機能(部品)として何が必要か？ ということを考えます．

まず，レンズ，ピント，ストロボ，オート・フォーカスなど，撮影する機能を実現する「写真機」というドメインが考えられます．また，ズーム，シャッタ，アクチュエータ，レンズ保護カバーなどの「光学系メカ」が考えられます．さらに，半押し，モード設定，ズーム操作，液晶表示部分などの「ユーザ・インターフェース」が考えられます．そして，JPEG(圧縮)，ディジタル・ズーム(拡大/縮小)，ノイズ除去，フィルタリング(ホワイト・バランス)などの「画像処理」，さらに，画像をメモリに保存する「ファイル・システム」が考えられます．

### ● 画像処理部分の分析モデルのシーケンス図を作成する

「画像を撮る」というシナリオの中でシャッタ・ボタンが押されたときの処理を考えてみましょう．ここでは画像処理部の分析モデルのシーケンス図を作成します(**図7**)．

最初に光学系メカから原画像データが送られてきます．画像補正クラスではホワイト・バランスなどのフィルタリング処理，ノイズ除去処理，拡大/縮小などのディジタル・ズーム処理が行われます．そして，補正された画像は画像加工クラスにおいてJPEG圧縮され，ファイル・システム

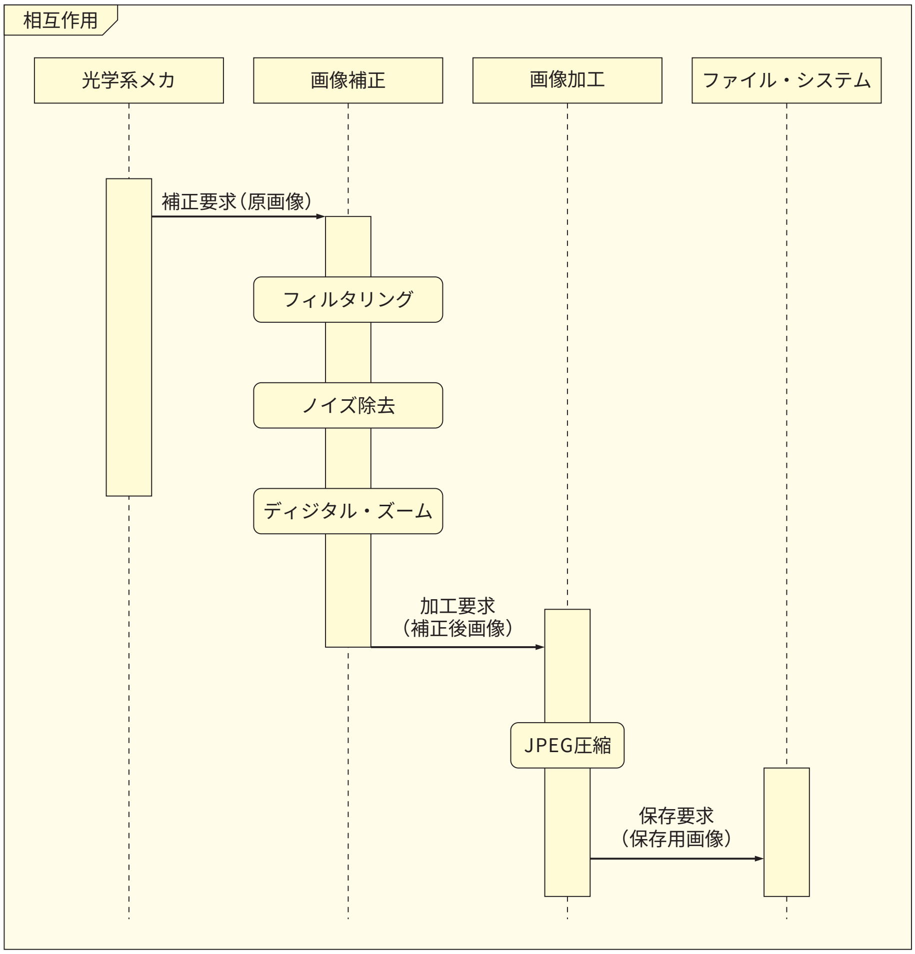



**図7　画像処理部分のシーケンス図**

シーケンス図では処理の流れを記述する．光学系メカから原画像データが送られてくる．画像補正クラスでフィルタリング処理，ノイズ除去処理，ディジタル・ズーム処理が行われる．そして，補正された画像は画像加工クラスでJPEG圧縮され，ファイル・システムに送られる．

に送られるといった流れになります．

### ● 画像処理部分の分析モデルのクラス図を作成する

上記のシーケンス図をもとに，クラス図を作成します(図8)．

まず，クラスを作成します．シーケンス図にもあるとおり，「光学系メカ・クラス」，「画像補正クラス」，「画像加工クラス」，「ファイル・システム・クラス」の四つのクラスを作成します．

次に，操作を考えてみましょう．画像補正クラスには，「フィルタリング」，「ノイズ除去」，「ディジタル・ズーム」などの操作があります．そして属性については，「フィルタリング画像」，「ノイズ除去画像」，「補正後画像」など，各操作を行う際にやりとりされる画像があります．これらは，外部からのアクセスを不可とします．そして，外部からのアクセスを可能にする「画像補正」という操作を作成します．外部からのアクセスは，すべてこの画像処理を返して行うことになります．そうすることにより，外部からの不正アクセスが減らせます．

また，画像加工クラスには，「JPEG圧縮」という操作を作成します．属性としては補正後画像を作成します．そして，同じように外部からのアクセスを不可とし，外部からのアクセスを可能にする「画像加工」という操作を作成します．

続いて，各クラスの関連を記述します．シーケンス図のメッセージがあるところに関連を記述します．

### ● 画像処理部分のSystemC構造図を作成する

最後に，画像処理部分のクラス図をSystemC構造図に変換します(図9)．

まずモジュールですが，光学系メカ(meka)，画像補正処理(Image_revision)，画像加工処理(Image_compression)，

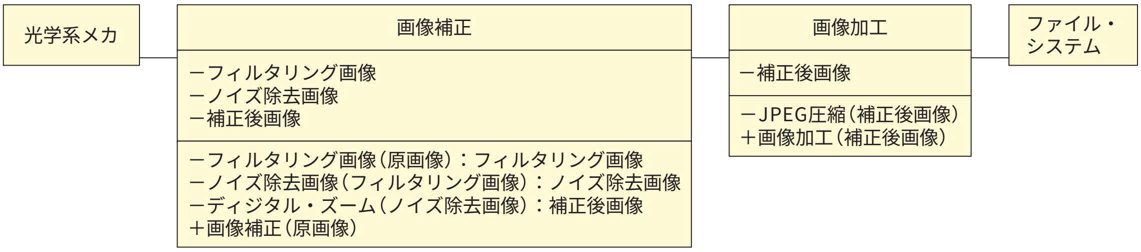



**図8　画像処理部分のクラス図**

**シーケンス図の処理を各クラスの記述とする．画像補正クラスには，操作として外部からアクセス可能な画像補正が存在する．内部の操作には，フィルタリングやノイズ除去，ディジタル・ズームなどがある．属性については，各操作が行われる際にやりとりされる画像がある．画像加工クラスにはJPEG圧縮という操作があり，属性として補正後画像がある．**

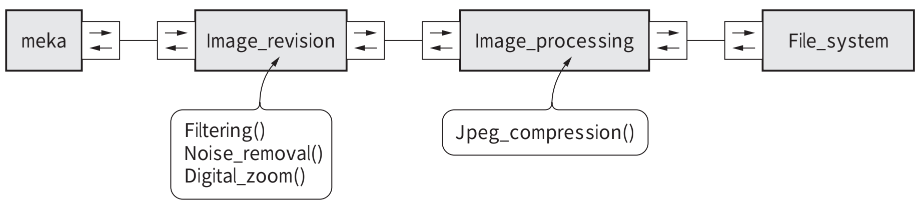



**図9**
**画像処理部分のSystemC構造図**

**画像処理部分のクラス図をSystemC構造図に変換した．光学系メカ(meka)，画像補正処理(Image_revision)，画像加工処理(Image_compression)，ファイル・システム(File_system)をポートで接続する．**

ファイル・システム(File_system)といった四つのモジュールが存在します．そして，各モジュールの間はポートで接続されています．また，フィルタリング(Filtering)，ノイズ除去(Noise_removal)，ディジタル・ズーム(Digital_Zoom)，JPEG圧縮(Jpeg_compression)などの操作はプロセスになります．

今回のモデル(各UML図)を見て疑問を感じる人もいるかもしれません．現在は，反復開発で言うとまだ1回目の反復の途中であり，モデルをこれからさらに洗練していくことになります．つまり，現状のモデルは最終成果物ではありません．

本稿では，ディジタル・カメラの画像処理部分に関して，シャッタ・ボタンが押された際のシナリオについてのみモデリングしました．UMTPでは，引き続きモデリング作業を進めます．洗練されたモデルについては，機会をあらためて報告していきたいと思います．

**つかだ・ゆういち**
**キャッツ(株)**
**http://www.zipc.com/**

**<筆者プロフィール>**
**塚田雄一．1989年に入社以来，キャッツひと筋．現在，SoC開発向け上流ツール「XModelink」の責任者であり，UMTP(UMLモデリング推進協議会)SoC分科会の主査．プライベートでは過去に車のレースにはまったこともあり，負けず嫌いで絶対優勝するまでやらないと気がすまない性格(よく子どもみたいなやつと言われる．家族(妻，娘)もいるのに…)である．**

# COLUMN UML，SystemC，モデリング関連の本誌記事一覧

Design Wave Magazine 編集部

ここではUMLやSystemC，システム・モデリング関連の情報が掲載された本誌の記事をご紹介いたします．

## ■UML関連記事■

**特集２●組み込み設計を効率化する『UML』**(全５章)
2003年２月号，pp.72-121／藤倉俊幸，杉浦英樹，人見繁，吉田寛，二上貴夫，山田大介，久我雅人 著

**総力特集●無償ツールで組み込み&半導体開発の全工程を体験**
**第３章　UMLによる組み込みシステム開発を体験する**
2003年４月号，pp.54-61／萩原裕志 著

**特集２●製品開発の生命線，ソフトウェア・テスト**
**第４章　UML 2.0で変わるコンポーネント・テスト**
2004年９月号，pp.56-62／伊藤昌夫 著

**特集２●組み込み設計＆LSI設計のためのUML**(全２章)
2004年12月号，pp.70-96／阿左美勝，綱嶋義人，小坂優，井山幸次，宮原忠義，岡田敏，難波睦，塚田雄一 著

**特集１●無償ツールでデバイス＆システム設計の全工程を体験**
**第４章　UMLとSystemCによるシステムLSI開発を体験する**
2005年４月号，pp.70-82／塚田雄一 著

## ■SystemC関連記事■

**特集１●「Cベース設計」にトライ！**(全５章)
2003年２月号，pp.20-70／藤田昌宏，長谷川裕恭，赤星博輝，平井慎一，山本真也，石井忠俊 著

**総力特集●無償ツールで組み込み&半導体開発の全工程を体験**
**第６章　SystemCによるLSI設計を体験する**
2003年４月号，pp.86-97／牧野潔 著

**連載●SystemCを使ったシステム設計**(全９回)
**第１回　SystemCの背景，インストール，プログラム実行**
2003年７月号，pp.115-122／長谷川裕恭 著
**第２回　データ・タイプ，モジュール，信号，ポート**
2003年８月号，pp.137-143／長谷川裕恭 著
**第３回　プロセス，アンタイムド，サイクル精度**
2003年９月号，pp.135-142／長谷川裕恭 著
**第４回　マスタ・スレーブ通信**
2003年11月号，pp.136-142／長谷川裕恭 著
**第５回　インターフェース接続**
2003年12月号，pp.126-135／長谷川裕恭 著
**第６回　C++の基礎知識**
2004年２月号，pp.132-138／長谷川裕恭，小川丈博 著
**第７回　続・C++の基礎知識**
2004年３月号，pp.107-114／長谷川裕恭，小川丈博 著
**第８回　プリミティブ・チャネルを使った記述**
2004年５月号，pp.139-147／長谷川裕恭，小川丈博 著
**第９回　sc_mutexと通信詳細化の記述**
2004年７月号，pp.123-129／長谷川裕恭，小川丈博 著

**特集２●Linuxをベースに設計環境を構築する**
**Appendix A**　SystemCシミュレータを使ってみよう
2004年７月号，pp.95-96／塚田雄一 著

**連載●SystemCによるビヘイビア合成入門**(全８回)
**第１回　SystemCの利点とビヘイビア合成を用いた設計フロー**
2004年11月号，pp.132-138／桜井至 著
**第２回　モジュールと入出力プロトコルの定義**
2004年12月号，pp.115-120／桜井至 著
**第３回　ビヘイビア合成の基本処理ステップ**
2005年２月号，pp.138-143／桜井至 著
**第４回　ビヘイビア合成可能なC/C++構文**
2005年３月号，pp.123-130／桜井至 著
**第５回　続・ビヘイビア合成可能なC/C++構文**
2005年５月号，pp.114-120／桜井至 著
**第６回　テストベンチと階層構造**
2005年６月号，pp.94-99／桜井至 著
**第７回　ビヘイビア合成のDCT設計への適用**
2005年７月号，pp.120-129／桜井至 著
**第８回　ビヘイビア設計の流れと今後の動向**
2005年９月号，pp.136-143／桜井至 著

## ■システム・モデリング関連記事■

**特集２●状態遷移図でシステム/ソフト/ハードをモデリング**(全３章)
2004年２月号，pp.83-118／山崎辰雄，小林芳直 著